世界モデルによる好奇心と新規性に基づく探索

脇聡志[1,a)]　鶴岡慶雅[2]

概要：強化学習では学習に必要なデータをエージェントが自ら収集するため, 未知の状態をどのように探索するかが重要である．探索の際, 好奇心や新規性を指標とする内発的報酬を与えることで学習を効率的に行えることが知られている．本研究では DreamerV2 と呼ばれる最新のモデルベース強化学習の手法を用いて好奇心ベースの内発的報酬を生成する．さらに新規性ベースの内発的報酬と組み合わせることで好奇心ベースの手法の多くで問題となる noisy-TV problem の解消を試みた．その結果, 探索が難しいとされる Montezuma's Revenge と Gravitar の環境下で提案手法がモデルフリー強化学習と内発的報酬を組み合わせた既存手法や DreamerV2 と好奇心ベースの内発的報酬を組み合わせた既存手法と比べ性能が優れた．また新規性ベースの内発的報酬と組み合わせることで noisy-TV problem を緩和できることを実験的に確かめた．

Exploration Driven by Curiosity and Novelty via World Models

Satoshi Waki[1,a)]　Yoshimasa Tsuruoka[2]

Abstract: In Reinforcement Learning, where the agent collects the data necessary for learning by itself, methods for exploring unknown states are important. An intrinsic reward based on curiosity and novelty makes it possible for an agent to learn efficiently during search. In this paper, we propose to generate curiosity-based intrinsic rewards using a state-of-the-art model-based reinforcement learning method called DreamerV2. We also attempted to solve the noisy-TV problem, which is a problem in many curiosity-based methods, by combining curiosity-based intrinsic rewards with novelty-based intrinsic rewards. As a result, our method outperformed previous methods that combined model-free reinforcement learning with intrinsic rewards or DreamerV2 with curiosity-based intrinsic rewards in Montezuma's Revenge and Gravitar environments, which are considered difficult to explore. We further experimentally confirmed that the noisy-TV problem can be solved by combining the novelty-based intrinsic reward.

1. はじめに

機械学習の手法の一つである強化学習はディープラーニングと組み合わせた深層強化学習により目覚ましい成果を上げている．その応用分野は囲碁や将棋といった完全情報ゲームから不完全情報ゲームへ, ゲーム攻略から現実世界の機械制御へと適応範囲を広げており, 強化学習の研究はますます重要になっている．

[1] 東京大学工学部電子情報工学科
Department of Information and Communication Engineering, The University of Tokyo
[2] 東京大学大学院情報理工学系研究科電子情報学専攻
Graduate School of Information Science and Technology, The University of Tokyo
[a)] waki@logos.t.u-tokyo.ac.jp

強化学習の課題の一つとして報酬が疎な環境下での探索がある．強化学習では環境から与えられる報酬に基づき方策と呼ばれる状態を入力とし行動を出力する関数を学習する．そのため方策の学習のためのフィードバックが乏しい報酬が疎な環境では効率的な探索が必要となる．探索を促す手段として確率 ε でランダムな探索をする ε-greedy やパラメータ空間にノイズを加える方法 [1, 2] が挙げられる．また他の手段として内発的報酬 (intrinsic reward) を導入する方法がある．内発的報酬は報酬が疎な環境下でもエージェントが環境を探索するよう内発的に動機付けるための報酬である．この内発的報酬には動機付けの仕方に基づいて主に好奇心ベースの方法と新規性ベースの方法の 2 種類に分類される．好奇心ベースの方法は環境のダイナミクス

を明らかにするような内発的報酬を与え, 新規性ベースの方法は過去の探索で訪れていない状態へ探索するよう内発的報酬を与える. 好奇心ベースの手法の多くには noisy-TV problem [3] と呼ばれる問題が指摘されており, noisy-TV problem を持たない好奇心ベースの手法や新規性ベースの手法がよく使われる.

近年では, 内発的報酬とモデルベース強化学習を組み合わせた手法が提案されている. モデルベース強化学習とは観測したデータから環境に関するモデルを構築し, そのモデルを利用し学習を行う強化学習を指す. 一方, 得たデータから直接学習する強化学習はモデルフリー強化学習と呼ばれる. 正確な環境のモデルを構築するのは困難であるため, これまで内発的報酬とモデルフリー強化学習を組み合わせた手法が多く提案されていた. しかし最近, 世界モデル (World Models) [4] といった機構が導入されることにより幾つかの環境下では長期予測が可能となるほど正確な環境のモデルが学習できるようになり, モデルベース強化学習の手法に内発的報酬を併せる試みも行われている [5, 6, 7].

従来の手法は内発的報酬の枠組みとモデルベース強化学習の枠組みを完全に分離してる. すなわち, モデルベース強化学習で学習した環境のモデルを一切利用せず内発的報酬を計算する [5, 6, 7]. それぞれの枠組みを分割することで余計に計算コストが必要となったり, 環境のモデルを持つモデルベース強化学習の利点を活かせない.

本研究の目的はモデルベース強化学習の利点を活かした性能の良い内発的報酬の設計をすることである. 貢献の一つ目は世界モデルを利用して好奇心の内発的報酬計算した点, 二つ目は noisy-TV problem を解決するために好奇心ベースの方法と新規性ベースの方法を組み合わせた点である. 実験の結果, 提案手法は従来の内発的報酬の手法と比べ効率的に探索でき, noisy-TV problem を緩和することが確かめられた.

本稿では第 2 章で関連研究の紹介, 第 3 章で提案手法の説明, 第 4 章で実験結果を示し, 第 5 章で結論を述べる.

2. 関連研究

2.1 好奇心ベースの内発的報酬

好奇心ベースの方法では現在の状態と行動から次状態を予測するモデルを学習し, そのモデルを洗練させるような状態や行動に対して報酬を与える [8, 9, 10]. 報酬を与える指標として予測結果と実際得られたデータの誤差を用いる手法がある. ICM (Intrinsic Curiosity Module)[11] はそのような手法の一つである. 予測誤差を指標とする手法には noisy-TV problem が指摘されている. Noisy-TV problem とは, 図 1 のように環境のダイナミクスが確率的な場合, 十分学習を行っても予測誤差が必ず発生し, 同じ領域を何度も探索する問題のことである. この noisy-TV problem の解決策として新規性ベースの方法や好奇心の指標として複数の予測モデルを学習し各予測結果の分散を用いる Disagreement と呼ばれる手法 [12] が提案されている.

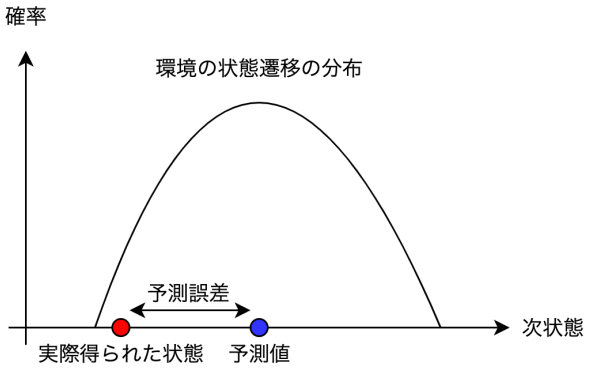


図 1: Noisy-TV problem の概念図. 環境の状態遷移が確率的な場合, モデルが十分学習され, 正しい期待値を出力していても必ず予測誤差が発生する. そのため好奇心の内発的報酬は 0 に収束せず同じ状態を何度も探索する.

2.2 新規性ベースの内発的報酬

新規性ベースの方法では新規性を測るためのアプローチとして各状態の訪問回数を数え, 訪問回数が少ない状態に対し内発的報酬を与えるカウントベースのアプローチ [13, 14, 15] や探索の各軌跡において連続する状態の表現が大きく異なる場合に内発的報酬を与える状態差分アプローチ [16], またそれらを組み合わせた手法がある [17, 18]. カウントベースのアプローチでは状態や行動が低次元の場合はそのまま出現回数を数えるが [19, 20], 高次元の場合は出現回数を近似した擬似カウント (pseudo-count) を用いる [21]. この擬似カウントとして Context Tree Switching density model (CTS 密度モデル) を用いて近似する手法 [22] や, PixelCNN を用いて近似する手法 [23], Random Network Distillation (RND) と呼ばれる近似手法 [3] がある. 状態差分アプローチでは状態表現の差を内発的報酬とすることで, 探索がよく行われている状態空間内の領域と行われていない領域の境界部を探索するよう促す.

2.2.1 RND

RND は簡易な実装と優れた性能からよく用いられる. RND ではニューラルネットワークを 2 つ用いることで擬似カウントを計算する. 一つ目のネットワークはランダムな重みをもつニューラルネットワークで, この出力値を二つ目のニューラルネットワークで予測する. このため前者をターゲットネットワーク, 後者を予測ネットワークという. ネットワーク間の誤差はデータの数とともに減少するため, RND では誤差の逆数を擬似カウントとして利用する. すなわち x_t を状態として $f : x_t \mapsto f(x_t) \in \mathbb{R}$ をターゲットネットワーク, $\hat{f}_\theta : x_t \mapsto \hat{f}_\theta(x_t) \in \mathbb{R}$ を予測ネットワークとすると内発的報酬として式 (1) のように与える.

$$r_t = \mathrm{RND}_\theta(x_t) \coloneqq \left(\hat{f}_\theta(x_t) - f(x_t)\right)^2 \qquad (1)$$

2.3 モデルベース強化学習

モデルベース強化学習は主に 3 つの過程で動作する．一つ目の過程では探索で得たデータから世界モデルと呼ばれる環境のダイナミクスを学習する．この世界モデルは通常, 状態を低次元にエンコードした潜在的な状態でダイナミクスを学習する [24, 25, 4]．世界モデルをコンパクトにすることで状態遷移を予測する計算量が削減できる．二つ目の過程はプランニングと呼ばれ, 学習した世界モデルを利用してマルチステップ予測を行い最適な行動を選択あるいは学習する．最後に三つ目の過程では最良の行動を実行し環境からデータを集める．

2.3.1 DreamerV2

DreamerV2 はモデルベース強化学習の手法の内, 初めて Atari のベンチマークで人間レベルの性能を達成した手法である．DreamerV2 の前身の Dreamer [26] では, プランニングでステップ間の逆伝搬を用いて方策や価値関数の学習を行う仕組みが導入された．それまでの手法では世界モデル上で複数の行動を実行し最良の行動を選択していたため計算量が多く近視眼的になる可能性があった．DreamerV2 では Dreamer に離散的な潜在空間を使用したり KL バランシングと呼ばれる手法を導入するなど修正し, 特に Atari のベンチマークの性能を向上させた．

DreamerV2 は環境の状態遷移のモデル化のために Recurrent State-Space Model (RSSM)[27] を用いた．RSSM は再帰型モデル, 状態表現モデル, 遷移予測モデルの 3 つのモデルから構成される．

$$\text{再帰型モデル：}\ h_t = f_\theta\left(h_{t-1}, z_{t-1}, a_{t-1}\right) \tag{2}$$

$$\text{状態表現モデル：}\ z_t \sim q_\theta\left(z_t \mid h_t, x_t\right) \tag{3}$$

$$\text{遷移予測モデル：}\ \hat{z}_t \sim p_\theta\left(\hat{z}_t \mid h_t\right) \tag{4}$$

ただし a_t が行動, x_t が環境の状態, z_t が x_t の潜在状態, h_t が再帰型モデルの隠れ状態である．遷移予測モデルの学習には式 (5) のように状態表現モデルとの KL 情報量 (Kullback–Leibler divergence) が用いられる．

$$\text{KL Loss} = \mathrm{KL}\left[q_\theta\left(z_t \mid h_t, x_t\right) \,||\, p_\theta\left(z_t \mid h_t\right)\right] \tag{5}$$

2.3.2 Plan2Explore

Plan2Explore[7] はモデルベース強化学習の Dreamer と好奇心ベースの内発的報酬を与える Disagreement を組み合わせた手法である．Plan2Explore では遷移予測モデルの入力を RSSM の隠れ状態 h_t と行動 a_t, 出力を次の状態表現 z_{t+1} として学習させモデル間での出力の分散を内発的報酬として与える．また一般的に内発的報酬の手法は環境からの報酬と内発的報酬の和を環境からの報酬とみなして学習するが, Plan2Explore は内発的報酬のみで探索を行う．

3. 提案手法

本研究ではモデルベース強化学習の手法として DreamerV2 を用い, DreamerV2 内の遷移予測モデルの学習に用いられる KL 情報量を予測誤差の指標とした．また新規性の指標として潜在状態と隠れ状態を入力とする RND で計算した擬似カウントを用いる．予測誤差を擬似カウントで割った値を内発的報酬として与える．

3.1 好奇心の内発的報酬

DreamerV2 では遷移予測モデルを状態表現モデルとの KL 情報量で学習するためこの KL 情報量の値が大きい程, 状態遷移の予測誤差が大きく, よい好奇心の指標となる．KL 情報量を計算するためには観測状態 x_t が必要であるが, 実際に内発的報酬を計算するプランニングの過程では潜在状態空間中で遷移させるため x_t は得られない．よって世界モデルを学習させるときに計算した KL 情報量を記憶するニューラルネットワーク $\widehat{\mathrm{KL}}_\vartheta : (z_t, h_t) \mapsto \widehat{\mathrm{KL}}_\vartheta\left(z_t, h_t\right) \in \mathbb{R}^{+\,*1}$ を用意した．KL 情報量は非負値であるため, 出力は非負実数である．ニューラルネットワークの詳細は付録 A.1 に示した．学習には以下の損失 $\mathcal{L}\left(\vartheta \mid \theta, \mathcal{D}_c\right)$ を用いた．ただし $\mathcal{D}_c$ はある時点 k から $k+L$ までの系列データ $\{(a_t, x_t, r_t, z_t, h_t)\}_{t=k}^{k+L}$ を表す．

$$\mathcal{L}\left(\vartheta \mid \theta, \mathcal{D}_c\right) \coloneqq \frac{1}{2L}\sum_{t=k}^{k+L}\left(\widehat{\mathrm{KL}}_\vartheta\left(z_t, h_t\right) - \mathrm{KL}\left[q_\theta\left(z_t \mid h_t, x_t\right) \,||\, p_\theta\left(z_t \mid h_t\right)\right]\right)^2 \tag{6}$$

3.2 新規性の内発的報酬

上記と同じ理由で観測状態 x_t を用いて内発的報酬を計算することが出来ないため RND の入力を潜在状態 z_t と隠れ状態 h_t を結合した $y_t \coloneqq z_t; h_t$ とした．すなわちターゲットネットワークを f, 予測ネットワークを $\hat{f}_\varphi$ としたとき $\mathrm{RND}_\varphi\left(y_t\right) = \left(f\left(y_t\right) - \hat{f}_\varphi\left(y_t\right)\right)^2$ を新規性の内発的報酬とする．ニューラルネットワークの詳細は付録 A.1 に示した．

3.3 全体像

RND の出力値は擬似カウントの逆数であるため, 2 つの内発的報酬の積を全体の内発的報酬とする．

$$r_t^{\text{intrinsic}}\left(z_t, h_t\right) \coloneqq \widehat{\mathrm{KL}}_\vartheta\left(z_t, h_t\right)\mathrm{RND}_\varphi\left(z_t; h_t\right) \tag{7}$$

積をとることで遷移が確率的な場合でも RND の出力値は減少するため全体の内発的報酬は 0 に収束し, noisy-TV problem が発生しないと期待される．

提案手法のアルゴリズムを Algorithm 1 に, 概要図を図 2 に示す．アルゴリズム中の下線部が DreamerV2 に付け加えた箇所である．DreamerV2 は一つ目の過程でデータ系列

*1 $\mathbb{R}^+ \coloneqq \{x \in \mathbb{R} \mid x \geq 0\}$

$\{(a_t, x_t, r_t)\}_{t=k}^{k+L} \sim \mathcal{D}$ をサンプリングする．その際，KL 情報量を記憶するモデル $\widehat{\mathrm{KL}}_\vartheta$ と状態の擬似カウントを計算するモデル RND_φ のパラメータを更新する．次に二つ目の過程でプランニングする際，式 (7) で計算した内発的報酬 $r_\tau^{\mathrm{intrinsic}}$ にスケール因子 α を掛け，環境からの報酬の予測値 $\hat{r}_\tau^{\mathrm{task}}$ を加えた値を用いて方策・状態価値関数を学習する．最後に三つ目の過程で学習した世界モデル・方策を基に探索しデータを集める．

Algorithm 1 提案手法

Ensure:
各モデルのパラメータ
世界モデル: θ, 方策: ϕ, 状態価値関数: ψ, $\widehat{\mathrm{KL}}$: ϑ, RND: φ.
ハイパーパラメータ
サンプリング間隔: C, サンプリングの系列長: L, 学習率: β, 予測ステップ数: H, 内発的報酬のスケール因子: α, 探索ステップ数の上限: T.

データセット $\mathcal{D}$ の初期化.
ニューラルネットワークのパラメータの初期化.
while not converged **do**
  for $c = 1$ **to** C **do**
    // 過程 1: 世界モデルの学習
    $\{(a_t, x_t, r_t)\}_{t=k}^{k+L} \sim \mathcal{D}$ をサンプリング.
    $\{(z_t, h_t)\}_{t=k}^{k+L}$ を計算.
    // $\mathcal{D}_c = \{(a_t, x_t, r_t, z_t, h_t)\}_{t=k}^{k+L}$ と置く.
    $\mathcal{D}_c$ から θ を更新.
    $\vartheta \leftarrow \vartheta - \beta \nabla_\vartheta \mathcal{L}(\vartheta \mid \theta, \mathcal{D}_c)$.
    $\varphi \leftarrow \varphi - \beta \nabla_\varphi \sum_t \mathrm{RND}_\varphi(z_t; h_t)$.
    // 過程 2: プランニング
    軌跡 $\{(z_\tau, h_\tau, a_\tau)\}_{\tau=t}^{t+H}$ を予測.
    報酬 $\hat{r}_\tau^{\mathrm{task}}(z_\tau, h_\tau)$ と状態価値 $v_\tau = v_\psi(z_\tau, h_\tau)$ の計算.
    式 (7) から $r_\tau^{\mathrm{intrinsic}}$ を計算.
    $\hat{r}_\tau \leftarrow \hat{r}_\tau^{\mathrm{task}} + \alpha\, r_\tau^{\mathrm{intrinsic}}$.
    // $\widehat{\mathcal{D}}_c = \{(z_\tau, h_\tau, a_\tau, \hat{r}_\tau, v_\tau)\}_{\tau=t}^{t+H}$ と置く.
    $\widehat{\mathcal{D}}_c$ から ϕ と ψ を更新.
  end for
  // 過程 3: 探索
  初期状態 x_0 にリセット.
  方策に基づき探索し $\mathcal{D}' = \{(a_t, x_t, r_t)\}_{t=0}^{T}$ を得る.
  $\mathcal{D} \leftarrow \mathcal{D} \bigcup \{\mathcal{D}'\}$.
end while

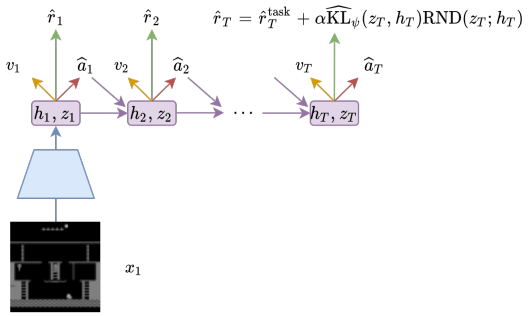


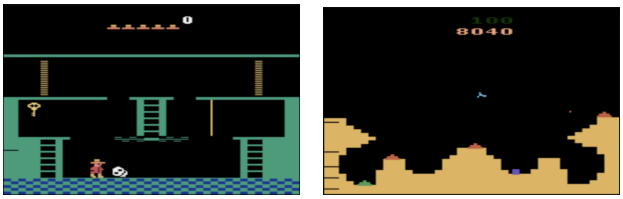
図 2: 提案手法の概要．プランニングの過程で環境から得られる報酬の予測値 $\hat{r}_t^{\mathrm{task}}$ に定数倍した内発的報酬を加えた値を基に探索を行う．

4. 実験

4.1 性能評価実験

4.1.1 実験設定

ベンチマークとして Atari に含まれるタスクの中で探索が難しいと言われている 7 つのタスク [28] の内，Montezuma's Revenge と Gravitar の 2 つの環境下で実験を行った．それぞれの環境では図 3 のようなゲーム画面が状態として与えられる．Montezuma's Revenge は複数の部屋を探索しゴールを目指すゲームである．Gravitar は宇宙船を操作し複数の惑星を訪れ各惑星表面に設置されている敵基地を破壊するゲームである．詳細な設定は Taïga らが行った実験設定 [28] と統一した．また性能を評価する際は，環境から得た報酬のみで学習した方策を用いた．ただし，このデータは探索で得たデータである．



(a) Montezuma's Revenge. **(b)** Gravitar.

図 3: 実験環境のゲーム画面.

4.1.2 比較手法

比較手法はモデルフリー強化学習の Rainbow [29] に CTS 密度モデルの内発的報酬を加えたモデル，PixelCNN の内発的報酬を加えたモデル，RND の内発的報酬を加えたモデル，ICM の内発的報酬を加えたモデル，ε-greedy で探索したモデル，パラメータにノイズを加えて探索する Noisy Networks (NoisyNets)[1] を適応したモデル，Plan2Explore で実験した．加えて，Montezuma's Revenge の環境下では提案手法内の RND による影響を確かめるため DreamerV2 に RND の内発的報酬のみを与えたモデルの性能も求めた．

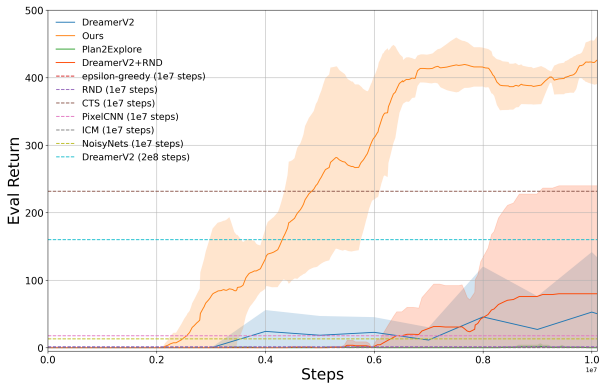
Rainbow に探索を促す機構を付け加えたモデルの性能は Taïga らの実験結果 [28] を引用した．また DreamerV2 の結果は Hafner らの実験結果 [30] を引用した．

Plan2Explore の実験では元の論文とは異なりモデルベース強化学習に DreamerV2 を用い，他の比較手法と条件を統一するために環境からの報酬と内発的報酬を加えた値で探索をするよう変更した．

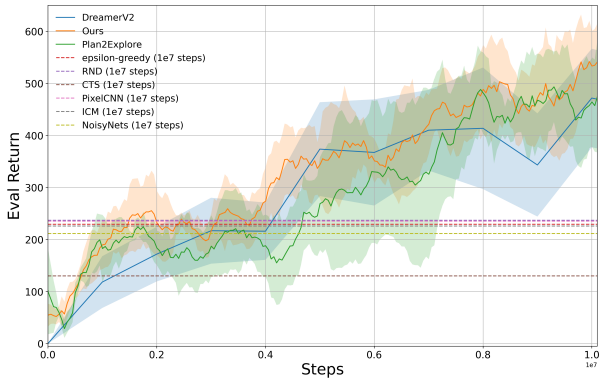
4.1.3 実験結果

Montezuma's Revenge の実験結果を図 4a に，Gravitar の実験結果を図 4b に示した．行動したステップ数に対する評価時の累積報酬を表している．DreamerV2, Plan2Explore, 提案手法 (Ours), 提案手法から好奇心の内発的報酬を除いたモデル (DreamerV2+RND) については複数のシードでの累積報酬の平均値と標準偏差を図示した．提案手法はど

ちらの環境下でもすべての比較手法より性能が優れた．さらに Montezuma's Revenge では DreamerV2 を 2 億ステップ学習させたときの性能を 1000 万ステップで上回った．また提案手法は DreamerV2+RND の性能を上回っていることから，提案手法が RND によってよい性能を示しているわけではないことが確認できる．



(a) Montezuma's Revenge での実験結果.



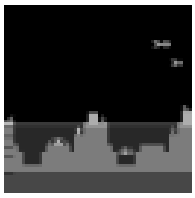
(b) Gravitar での実験結果.

図 4: 各環境での実験結果．横軸が行動したステップ数，縦軸が学習したモデルで評価した時の累積報酬である．移動平均 $r'_t = \sum_{i=-4}^{4} r_{t+i}/9$ を用いて平滑化した．

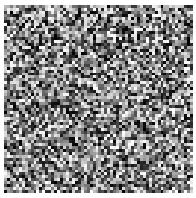
4.2 Noisy-TV problem に対する検証実験

4.2.1 環境設定

Noisy-TV problem に対する実験を Atari の中で簡単なタスクである Atlantis で行った．Atlantis は 3 つの大砲を操作して上空を飛ぶ宇宙船を落とすゲームである．環境にノイズを引き起こす行動 (Noisy action) を追加し，その行動を選ぶと図 5b のようにゲーム画面全面が 1 フレームだけノイズになるよう環境を変更し実験した．この時ノイズの種類を 10 種類に設定した．実験は好奇心の内発的報酬と新規性の内発的報酬を組み合わせた提案手法と好奇心の内発的報酬のみで探索する手法 (Curiosity only) をそれぞれもとの環境とノイズを加えた環境で行った．性能を評価する際は，環境から得た報酬のみで学習した方策を用いた．ただし，このデータは探索で得たデータである．



(a) ノイズがない時の画像.

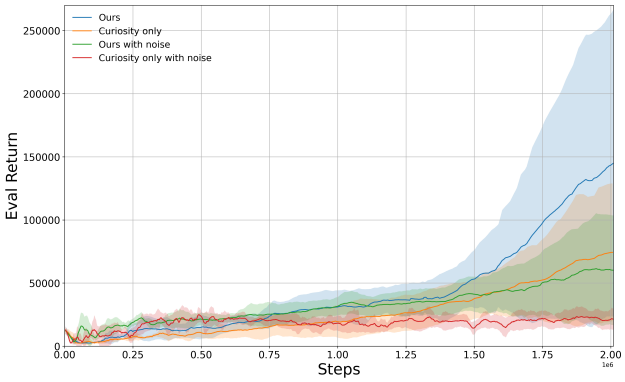


(b) ノイズがある時の画像.

図 5: エージェントが環境から得る画像.

4.2.2 実験結果

図 6a と図 6b に実験結果を示した．図 6a は各環境下，各手法の累積報酬の遷移を表している．ノイズを加えた環境で，提案手法は好奇心の内発的報酬のみで探索した手法の性能を上回った．また好奇心のみで学習した手法は 50 万ステップ後から性能改善が見られないのに対し，提案手法は 200 万ステップ時も性能が向上している．図 6b は探索時，全行動に対して，ノイズを引き起こす行動を選んだ割合を表している．どちらも学習開始直後に割合が急激に増え，緩やかに減少後，一定値に収束している．199 万ステップから 200 万ステップの間で平均値を計算すると提案手法は 15% に対し，好奇心のみで学習した手法は 78% であり，最終的な割合は 5.2 分の 1 になった．よって新規性の内発的報酬を組み合わせることで noisy-TV problem が緩和できることが実証された．



(a) 横軸が行動したステップ数，縦軸が学習したモデルで評価した時の累積報酬である.

(b) 横軸が行動したステップ数，縦軸が探索時にノイズを引き起こす行動を選んだ割合である.

図 6: Atlantis での noisy-TV problem に対する実験結果． それぞれ移動平均 $y'_t = \sum_{i=-4}^{4} y_{t+i}/9$ を用いて平滑化した.

5. おわりに

モデルベース強化学習の手法内の遷移予測モデルを利用して内発的報酬を計算する提案手法は, 報酬が疎な環境下で既存手法の性能を上回った. さらに新規性ベースの内発的報酬と組み合わせることで noisy-TV problem を緩和できることを実験で示した.

参考文献

[1] Meire Fortunato et al. Noisy Networks for Exploration. In *ICLR*, 2018.

[2] Matthias Plappert et al. Parameter Space Noise for Exploration. In *ICLR*, 2018.

[3] Yuri Burda et al. Exploration by Random Network Distillation. In *ICLR*, 2019.

[4] David R Ha and Jürgen Schmidhuber. World Models, 2018.

[5] Younggyo Seo et al. State Entropy Maximization with Random Encoders for Efficient Exploration. In *ICML*, 2021.

[6] Tianjun Zhang et al. MADE: Exploration via Maximizing Deviation from Explored Regions. *ArXiv*, Vol. abs/2106.10268, , 2021.

[7] Ramanan Sekar et al. Planning to Explore via Self-Supervised World Models, 2020.

[8] Pierre-Yves Oudeyer. Computational Theories of Curiosity-Driven Learning, 2018.

[9] Pierre-Yves Oudeyer, F. Kaplan, and V. Hafner. Intrinsic Motivation Systems for Autonomous Mental Development. *IEEE Transactions on Evolutionary Computation*, Vol. 11, pp. 265–286, 2007.

[10] Nick Haber et al. Learning to Play with Intrinsically-Motivated Self-Aware Agents. In *NeurIPS*, 2018.

[11] Deepak Pathak et al. Curiosity-Driven Exploration by Self-Supervised Prediction. *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 488–489, 2017.

[12] Deepak Pathak et al. Self-Supervised Exploration via Disagreement. In *ICML*, pp. 5062–5071, 2019.

[13] Sebastian B. Thrun. Efficient Exploration In Reinforcement Learning. Technical report, USA, 1992.

[14] M. Sato, K. Abe, and H. Takeda. Learning control of finite Markov chains with an explicit trade-off between estimation and control. *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 18, No. 5, pp. 677–684, 1988.

[15] A. Barto and Satinder Singh. On the Computational Economics of Reinforcement Learning. 1991.

[16] Jingwei Zhang et al. Scheduled Intrinsic Drive: A Hierarchical Take on Intrinsically Motivated Exploration, 2019.

[17] Tianjun Zhang et al. BeBold: Exploration Beyond the Boundary of Explored Regions, 2020.

[18] Roberta Raileanu and Tim Rocktäschel. RIDE: Rewarding Impact-Driven Exploration for Procedurally-Generated Environments, 2020.

[19] Alexander L. Strehl and Michael L. Littman. An analysis of model-based Interval Estimation for Markov Decision Processes. *J. Comput. Syst. Sci.*, Vol. 74, pp. 1309–1331, 2008.

[20] J. Z. Kolter and A. Ng. Near-Bayesian exploration in polynomial time. In *ICML*, 2009.

[21] M. Lopes et al. Exploration in Model-based Reinforcement Learning by Empirically Estimating Learning Progress. In *NIPS*, 2012.

[22] Marc G. Bellemare et al. Unifying Count-Based Exploration and Intrinsic Motivation. In *NIPS*, 2016.

[23] Georg Ostrovski et al. Count-Based Exploration with Neural Density Models, 2017.

[24] Manuel Watter et al. Embed to Control: A Locally Linear Latent Dynamics Model for Control from Raw Images. In *NIPS*, 2015.

[25] Maximilian Karl et al. Deep Variational Bayes Filters: Unsupervised Learning of State Space Models from Raw Data. In *ICLR*, 2017.

[26] Danijar Hafner et al. Dream to Control: Learning Behaviors by Latent Imagination. In *ICLR*, 2020.

[27] Danijar Hafner et al. Learning Latent Dynamics for Planning from Pixels. In *ICML*, pp. 2555–2565, 2019.

[28] Adrien Ali Taiga et al. On Bonus Based Exploration Methods In The Arcade Learning Environment. In *ICLR*, 2020.

[29] Matteo Hessel et al. Rainbow: Combining Improvements in Deep Reinforcement Learning. In *AAAI*, 2018.

[30] Danijar Hafner et al. Mastering Atari with Discrete World Models. In *ICLR*, 2021.

付　　録

A.1 ハイパーパラメータ

実験結果はすべて 5 つのシードで実験した時の平均・標準偏差である. また下記のモデルの最適化には学習率 10^{-5}, Weight decay 10^{-6}, Gradient clipping 100 の Adam を用

いた.

DreamerV2 に関するハイパーパラメータの変更点として, 学習開始時にランダムに集めるデータサイズを 100 に, 訓練時に行動に加えるノイズの量を 0 にした.

スケール因子のハイパーパラメータ調整に関して Taïga ら [28] に倣い, Montezuma's Revenge でハイパーパラメータ調節をし, Gravitar ではその調節された値で実験を行った.

A.1.1 Plan2Explore

Plan2Explore は予測モデルにユニット数 400 の層を 4 層全結合し, ELU (Exponential Linear Unit) を活性化関数に用い, 10 個のモデル間での分散を計算した.

スケール因子 α は $\{0.0001, 0.001, 0.01, 0.1, 1, 10, 100, 1000\}$ の中で 0.0001 が最も性能が良かった.

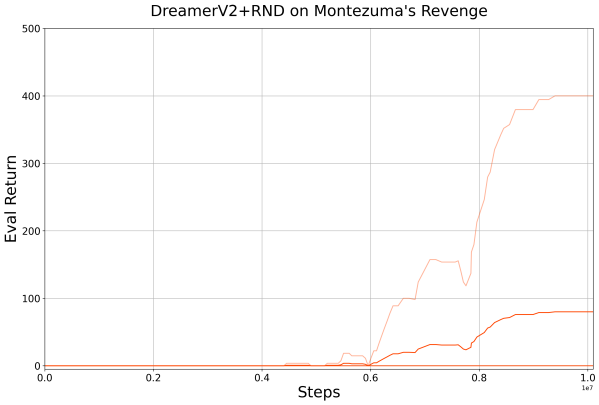
A.1.2 提案手法

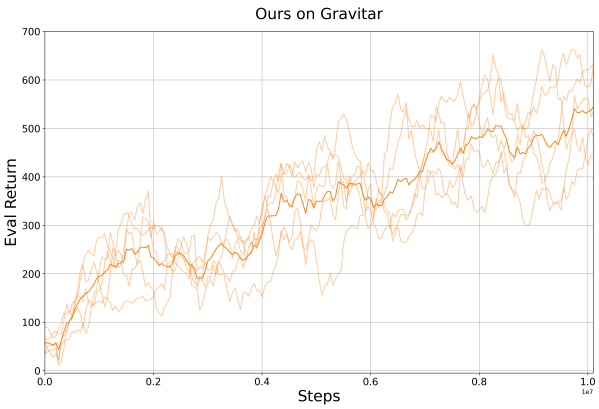
好奇心の内発的報酬を求める $\widehat{\mathrm{KL}}_\vartheta$ はユニット数 400 の層を 4 層全結合し, ELU を活性化関数に用いた. また最後の活性化関数には絶対値を追加した. 学習していない入力に対して大きな値を出力するようニューラルネットワークの初期の重みは平均 0.01, 標準偏差 0.05 の正規分布に従うようにした.

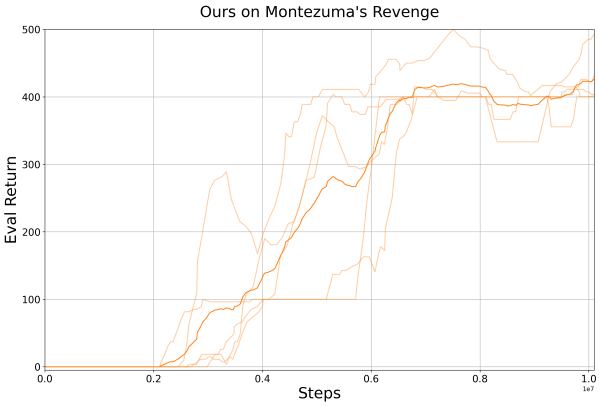
新規性の内発的報酬 RND_φ 内の予測ネットワークとターゲットネットワークはユニット数を順に 50, 10 の順に設定し全結合で繋げた. 活性化関数はすべて ELU にした.

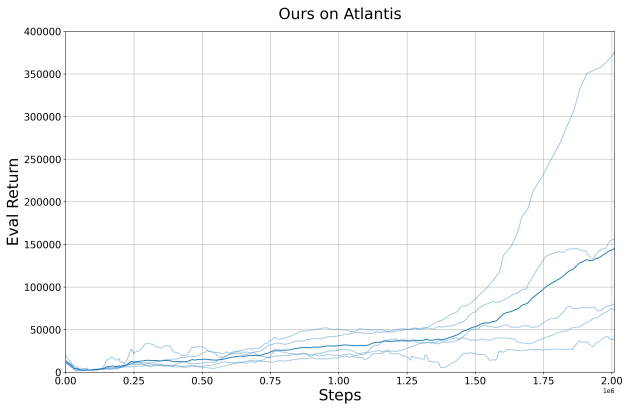
スケール因子 α は $\{0.001, 0.01, 0.1, 1, 10, 100\}$ の中で 0.1 が最も性能が良かった. また Montezuma's Revenge での DreamerV2+RND の実験は新規性の影響を調べる実験であったため, 内発的報酬のスケールが提案手法と同じになるようスケール因子として 0.06 を用いた.

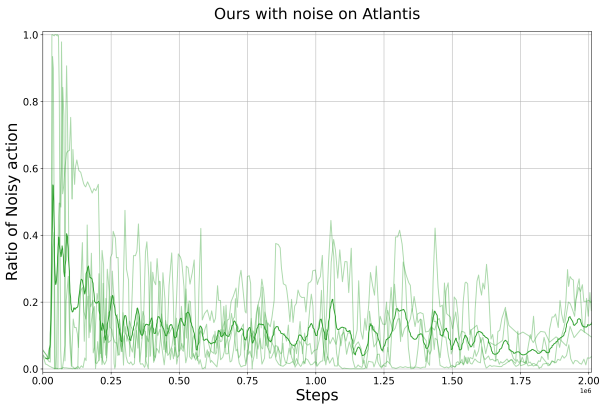
A.2 追加の図

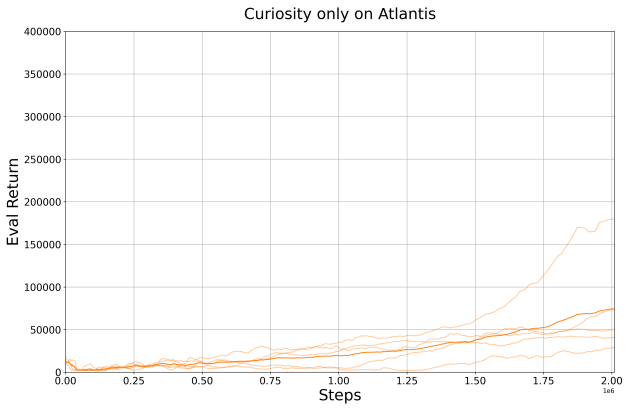












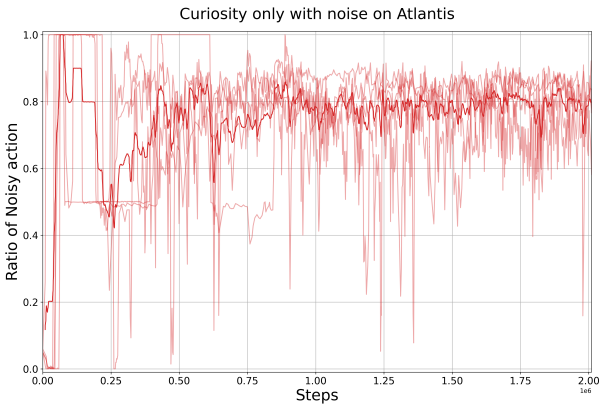


図 A·1: 各シード値での遷移．太線が平均値である．移動平均 $y'_t = \sum_{i=-4}^{4} y_{t+i}/9$ を用い平滑化した．